

JP-A 11-53049

(54) [Title of the Invention]      Computer System

(57) [Abstract]

[Problems to be Solved]

To allow controlling the stop/restart of a bus clock without mounting special functions onto peripheral devices on a bus, and to reduce the power consumption of a computer.

[Means for Solving Problems]

If a bus monitoring circuit 161 detects that both FRAME# and IRDY# are de-asserted and a bus request detection circuit 162 and a system event detection circuit 163 detect that a bus request signal and a system event signal are not generated, respectively, then an output of a 3-input OR gate 164 turns into "L" level indicating that the bus is not operating. As a result, a clock control signal output circuit 165 generates a clock control signal indicating the stop of PCICLK, whereby supply of PCICLK to respective PCI devices is stopped.

[What is Claimed is]

[Claim 1] A computer system characterized by comprising:

a plurality of peripheral devices coupled to a bus of the computer system;

bus clock generation means for generating a bus clock to give timing of a transaction on said bus to the plurality of peripheral devices;

bus idle detection means for monitoring the transaction on said bus, and for detecting whether or not a state is a bus idle state;

event detection means for detecting whether or not a bus request signal and an interrupt signal are generated from each of said plurality of peripheral devices;

clock control means for controlling a clock generating operation of said bus clock generation means based on detection results of said bus idle detection means and said event detection means, the clock control means stopping said bus clock if the bus idle state is detected and it is detected that said bus request signal and said interrupt signal are not generated.

[Claim 2] A computer system according to claim 1, characterized in that

said clock control means comprises:

means for delaying timing for stopping said bus clock so that said bus clock is stopped a certain period of time after said bus idle state is detected and it is detected that said bus request signal and said interrupt signal are not generated; and

means for restarting supply of said bus clock when it is detected that said bus idle state is released and that said bus request signal or said interrupt signal is generated.

[Claim 3] A computer system characterized by comprising:

a plurality of peripheral devices coupled to a bus of the computer system;

bus clock generation means for generating a bus clock to give timing of a transaction on said bus to the plurality of devices;

bus idle detection means for monitoring the transaction

on said bus, and for detecting whether or not a state is a bus idle state;

event detection means for detecting whether or not a bus request signal and an interrupt signal are generated from each of said devices; and

clock control means for controlling a clock generating operation of said bus clock generation means based on detection results of said bus idle detection means and said event detection means, said clock control means decreasing a frequency of said bus clock when the bus idle state is detected and it is detected that said bus request signal and said interrupt signal are not generated.

[Claim 4] A computer system according to claim 3, characterized in that

said clock control means comprises:

means for delaying timing for decreasing the frequency of said bus clock so that the frequency of said bus clock is decreased a certain period of time after said bus idle state is detected and it is detected that said bus request signal and said interrupt signal are not generated; and

means for returning the frequency of said bus clock to an original state when it is detected that said bus idle state is released and that said bus request signal or said interrupt signal is generated.

[Detailed Description of the Invention]

[0001]

[Field of the Invention]

This invention relates to a computer system, and particularly relates to a computer system having a bus clock control function.

[0002]

[Prior Art]

As system buses used for personal computers, an ISA (Industry Standard Architecture) bus and an EISA (Extended ISA) bus are conventionally, mainly used. In recent years, because of the acceleration of data transfer rate and the construction of a system architecture which does not rely on a processor, a PCI (Peripheral Component Interconnect) bus is gradually adopted.

[0003]

In case of the PCI bus, almost all data transfer is basically block transfer and the block transfer is realized by burst transfer. It is thereby possible to realize a data transfer rate of a maximum of 133 megabytes/second (when a data bus is 32 bits in width) on the PCI bus.

[0004]

Accordingly, if the PCI bus is adopted, data transfer between I/O devices and that between a system memory and an I/O device can be accelerated and system performance can be thereby improved.

[0005]

In the PCI bus architecture stated above, a relatively fast bus clock of 33MHz<sub>2</sub> is employed so as to realize fast operation. This bus clock is intended to provide the timing

of a transaction on the PCI bus to respective peripheral devices on the bus. While the system is being operating, the bus clock is continuously supplied to the peripheral devices on the bus.

[0006]

[Problems that the Invention is to Solve]

Such a fast bus clock is, however, one of large factors to increase the power consumption of a computer. For that reason, a specification for stopping the bus clock ("CLKRUN" protocol) has been incorporated into the specifications of the PCI bus.

[0007]

To mount this "CLKRUN" protocol on the system, however, it is necessary that all the devices on the PCI bus have functions corresponding to the "CLKRUN" protocol. Due to this, if there exists a peripheral device which does not correspond to the "CLKRUN" protocol, the "CLKRUN" protocol cannot be used, with the result that the bus clock cannot be stopped.

[0008]

This invention has been made in view of the above-stated respects. It is an object of this invention to provide a computer system capable of controlling a bus clock without mounting special functions such as a "CLKRUN" protocol onto peripheral devices and capable of easily reducing power consumption.

[0009]

[Means for Solving the Problems]

A computer system according to this invention is

characterized by comprising: a plurality of peripheral devices coupled to a bus of the computer system; bus clock generation means for generating a bus clock to give timing of a transaction on said bus to the plurality of peripheral devices; bus idle detection means for monitoring the transaction on said bus, and for detecting whether or not a state is a bus idle state; event detection means for detecting whether or not a bus request signal and an interrupt signal are generated from each of said plurality of peripheral devices; clock control means for controlling a clock generating operation of said bus clock generation means based on detection results of said bus idle detection means and said event detection means, the clock control means stopping said bus clock if the bus idle state is detected and it is detected that said bus request signal and said interrupt signal are not generated.

[0010]

In this computer system, by detecting a bus idle state and the presence/absence of a bus request signal and an interrupt signal, the state of the system can be examined. If the bus idle state is detected and it is detected that a bus request signal and an interrupt signal are not generated, then it is judged that the system is idle and the bus clock is stopped. In this case, as long as a bus request signal or an interrupt signal is generated even in the bus idle state, the bus clock is not stopped. Due to this, the operations of the peripheral devices are not influenced. It is, therefore, possible to stop an unnecessary bus clock without mounting special functions

such as a "CLKRUN" protocol onto the peripheral devices and thereby reduce power consumption.

[0011]

Further, it is preferable that said clock control means comprises means for delaying timing for stopping said bus clock so that said bus clock is stopped a certain period of time after said bus idle state is detected and it is detected that said bus request signal and said interrupt signal are not generated; and means for restarting supply of said bus clock when it is detected that said bus idle state is released and that said bus request signal or said interrupt signal is generated.

[0012]

When the bus clock is promptly stopped in the system idle state, it takes time before the supply of the bus clock is restarted even if an interrupt signal or the like is generated after the stop of the bus clock, thus causing the deterioration of system performance. To prevent the deterioration of the system performance, therefore, it is necessary to execute as many processings as possible collectively within a clock supply period. This can be realized by delaying timing for stopping the bus clock.

[0013]

Moreover, a computer system according to this invention is characterized by comprising: a plurality of peripheral devices coupled to a bus of the computer system; bus clock generation means for generating a bus clock to give timing of a transaction on said bus to the plurality of devices; bus idle

detection means for monitoring the transaction on said bus, and for detecting whether or not a state is a bus idle state; event detection means for detecting whether or not a bus request signal and an interrupt signal are generated from each of said devices; and clock control means for controlling a clock generating operation of said bus clock generation means based on detection results of said bus idle detection means and said event detection means, said clock control means decreasing a frequency of said bus clock when the bus idle state is detected and it is detected that said bus request signal and said interrupt signal are not generated.

[0014]

According to this constitution, the frequency of the bus clock is decreased instead of completely stopping the bus clock. The peripheral devices are normally constituted so that part of logics in the devices can operate even while the bus clock is being stopped and an interrupt signal and a bus request signal can be normally generated, so that no problem arises. However, some peripheral devices cannot operate at all if the bus clock is stopped. If such a device is connected, it is necessary to supply a minimum clock and to ensure the operation of such a device without completely stopping the bus clock. By doing so, it is possible to reduce power consumption while ensuring device operation.

[0015]

[Embodiment of the Invention]

The embodiment of this invention will be described



hereinafter with reference to drawings. FIG. 1 shows the constitution of a computer system according to one embodiment of this invention. This computer system is a notebook or laptop type portable computer which can be driven by batteries. On the system board of this computer system, there are provided a processor bus 1, a PCI bus 2, an ISA bus 3, a CPU 11, a host-PCI bridge device 12, a main memory 13, various types of PCI peripheral devices 14 and 15, a bus clock control circuit 16, a bus clock generation circuit 17, a PCI-ISA bridge device 18 and various types of ISA peripheral devices 19 and 20 and the like.

[0016]

Among these components, those functioning as PCI devices connected to the PCI bus 2, i.e., the host-PCI bridge device 12, the various types of PCI peripheral devices 14 and 15, and the PCI-ISA bridge device 18 are supplied with a PCI bus clock (PCICLK) generated by the bus clock generation circuit 17. This PCI bus clock (PCICLK) is intended to give the timing of a bus transaction to the respective PCI devices. All of cycles on the PCI bus 2 are executed synchronously with the PCI bus clock (PCICLK).

[0017]

Next, description will be given to the functions and constitutions of the respective components shown in FIG. 1. The CPU 11 is realized by, for example, a microprocessor "Pentium" manufactured and sold by Intel Corp. or the like. The processor bus 1 directly coupled to the input and output pins of this CPU

11 has a data bus of 64 bits in width. The main memory 13 is a memory device storing an operating system, a device driver, an application program to be executed, processing data and the like. The main memory 13 consists of a plurality of DRAM modules.

[0018]

The host-PCI bridge device 12 is a bridge LSI connecting the processor bus 1 to the PCI bus 2 and functions as one of the bus masters of the PCI bus 2. This host-PCI bridge device 12 has a function of two-way converting bus cycles including data and addresses between the processor bus 1 and the PCI bus 2, a function of controlling access to the main memory 13 through the memory bus and the like.

[0019]

The PCI bus 2 is a clock synchronous type input and output bus and all cycles on the PCI bus 2 are executed synchronously with the PCI bus clock as stated above. The maximum frequency of the PCI bus clock is 33 MHz. The PCI bus 2 has an address/data bus used in a time division manner. This address/data bus is 32 bits in width.

[0020]

A data transfer cycle on the PCI bus 2 consists of an address phase and one or more data phases following the address phase. In the address phase, an address and a transfer type are outputted. In the data phases, data of 8 bits, 16 bits, 24 bits or 32 bits is outputted.

[0021]

The PCI devices 14 and 15 are, for example, a graphics controller, a PC card (or card bus) controller, an IrDA controller, an SCSI controller and/or the like. Each of the PCI devices 14 and 15 functions as the bus master of the PCI bus 2 as in the case of the host-PCI bridge device 12.

[0022]

The PCI-ISA bridge device 16 is a bridge LSI connecting the PCI bus 2 to the ISA bus 3 and functions as one of the PCI devices. The ISA devices 19 and 20 on the ISA bus 3 are, for example, an HDD, a system timer, a keyboard controller and/or the like.

[0023]

The bus clock control circuit 16 controls the stop/restart of the supply of the PCI bus clock (PCICLK) to the respective PCI devices. The control of the circuit 16 over the stop/restart of the PCI bus clock (PCICLK) is conducted based on a bus idle signal, a bus request signal and a system event signal. The bus idle signal indicates the operating state of the PCI bus 2, i.e., whether the PCI bus 2 operates (or executes a bus cycle). The bus request signal is a signal with which a bus master requests the right of using the PCI bus 2 to a PCI bus arbiter. The system event signal indicates that some event such as an interrupt signal, occurs in the system.

[0024]

The bus clock control circuit 16 detects that the PCI bus is idle using the bus idle signal. If the bus request signal and the system event signal are not generated, the bus clock

control/circuit 16 controls the bus clock generation circuit 17 using the clock control signal to stop the bus clock. Also, if a bus request signal or a system event signal occurs while the bus clock is being stopped, the bus clock control circuit 16 controls the bus clock generation circuit 17 using the clock control signal to restart the supply of the bus clock.

[0025]

FIG. 2 shows an example of the concrete constitution of the bus clock control circuit 16. As shown in FIG. 2, the bus clock control circuit 16 consists of a bus monitoring circuit 161, a bus request detection circuit 162, a system event detection circuit 163, an OR gate 164 and a clock control signal output circuit 165.

[0026]

The bus monitoring circuit 161, which is intended to generate the above-stated bus idle signal, monitors a transaction on the PCI bus 2 using a frame signal FRAME# and an initiator ready signal IRDY# defined on the PCI bus 2, asserts the bus idle signal at "H" level when it is detected that the bus is not operating and de-asserts the bus idle signal at "L" level when it is detected that the bus is operating.

[0027]

Here, FRAME# is a signal driven by a present master to indicate the start of a transaction and the period of the transaction. When the signal FRAME# is de-asserted, it indicates that the transaction is in the final data phase. IRDY# is a signal driven by the present master and asserted to

show that the master has outputted defined data to the bus in a write cycle and asserted to show that the master is ready to receive data in a read cycle.

[0028]

When both FRAME# and IRDY# are de-asserted, the bus monitoring circuit 161 detects that the bus is idle and asserts the bus idle signal at "H" level.

[0029]

The bus request detection circuit 162 monitors all bus request signals REQ1# to REQn# inputted from the respective PCI devices to the PCI bus arbiter and asserts the bus request signal at "H" level if one of the request signals REQ1# to REQn# is asserted.

[0030]

The system event detection circuit monitors an interrupt signal INTA-D inputted from the respective PCI devices to the interrupt controller, an interrupt signal IRQ0-15 inputted from the respective ISA devices to the interrupt controller, a non-maskable interrupt signal NMI and a system management interrupt signal SMI. If one of the interrupt signals occurs, the system event detection circuit asserts that the system event signal is at "H" level.

[0031]

The bus idle signal from the bus monitoring circuit 161 is inversely inputted to the first input of the 3-input OR gate 164. Also, the bus request signal and the system event signal from the bus request detection circuit 162 and the system event

detection circuit 163 are inputted to the second input and the third input of the 3-input OR gate 164 as they are, respectively. The output of the 3-input OR gate 164 indicates that the bus is operating ("H" level)/the bus is not operating ("L" level). The output of the 3-input OR gate 164 is inputted to the clock control signal output circuit 165.

[0032]

The clock control signal output circuit 165 generates a clock control signal indicating the stop/restart of PCICLK based on the output of the 3-input OR gate 164. Next, the operation of this embodiment will be described with reference to FIGS. 3 and 4.

[0033]

A timing chart shown in FIG. 3 shows timing for stopping the bus clock (PCICLK). Namely, if it is detected that both RRAME# and IRDY# are de-asserted and that the bus request signal and the system event signal are not generated, the output of the 3-input OR gate 164 turns into "L" level indicating that the bus is not operating. As a result, the clock control signal output circuit 165 generates a clock control signal indicating the stop of PCICLK, whereby the supply of PCICLK to each of the PCI devices is stopped.

[0034]

If a clock request signal is generated (i.e., if the bus monitoring circuit 161 detects the assertion of FRAME# and IRDY#, the bus request detection circuit 162 detects the occurrence of a bus request signal or the system event detection circuit

164 detects the occurrence of an interrupt signal) while the supply of PCICLK is being stopped, the output of the 3-input OR gate 164 turns into "H" level indicating that the bus is operating. As a result, the clock control signal output circuit 165 generates a clock control signal indicating the restart of PCICLK, whereby the supply of PCICLK to each of the PCI devices is restarted.

[0035]

As can be seen, according to the constitution shown in FIG. 2, a system state can be examined by detecting whether the bus is idle and detecting whether or not a bus request signal and an interrupt signal are generated. If it is detected that the bus is in an idle state and that a bus request signal and an interrupt signal are not generated, then it is judged that the system is idle and PCKCLK is stopped. In this case, as long as a bus request signal or an interrupt signal is generated even while the bus is idle, PCICLK is not stopped, so that the operations of the respective PCI devices are not influenced at all. Further, if a bus request signal or an interrupt signal is generated from a certain PCI bus while PCICLK is being stopped, the supply of PCICLK is restarted and it is possible to carry out a normal transaction synchronous with PCICLK. It is, therefore, possible to stop an unnecessary bus clock without mounting special functions such as a "CLKRUN" protocol onto the PCI devices and to thereby reduce power consumption.

[0036]

FIG. 5 shows the second example of the constitution of

the bus clock control circuit 16. Here, a snap timer 166 is provided besides the constitution of FIG. 2. This snap timer 166 is intended to delay timing for stopping PCICLK by a certain period of time. When the output of the 3-input OR gate 164 turns into "L" level indicating that the bus is not operating, the snap timer 166 starts a count operation and if the count value reaches a predetermined value, the timer 166 notifies the clock control signal output circuit 165 that the count value reaches the predetermined value. As a result, the generation of a clock control signal indicating the stop of PCICLK is delayed by as much as the count time of the snap timer 166. The count value of the snap timer 166 can be made programmable by, for example, providing a configuration register for setting counter values in the bus clock control circuit 166 and setting a counter value corresponding to desired delay time by means of a software.

[0037]

FIG. 6 shows a manner of the transition of the state of the bus clock control circuit 16 shown in FIG. 15. In FIG. 6, a state S1 (RUN) shows a state in which PCICLK is supplied. If it is detected that the bus is not operating in this state, the state is moved to a state S2 (SNAP). In the state S2 (SNAP), the snap timer 166 carries out a count operation. If the generation of an interrupt signal or that of a bus request signal is detected while the snap timer 166 is carrying out the count operation, the state is returned to the state S1 (RUN) and the count value of the snap timer 166 is returned to an initial value. On the other hand, if the snap timer 166 completes the count



operation in the state S2 (SNAP), the state is moved to a state S3 (STOP). In the state S3 (STOP), the supply of PCICLK is stopped. If the generation of an interrupt signal or that of a bus request signal is detected in the state S3 (STOP), the state is returned to the state S1 (RUN).

[0038]

FIG. 7 shows operation timing when the state S1 (RUN) is changed to the state S3 (STOP) by way of the state S2 (SNAP). That is, if it is detected that both FRAME# and IRDY# are de-asserted and that a bus request signal and a system event signal are not generated, then the output of the 3-input OR gate 164 turns into "L" level indicating that the bus is not operating. As a result, the snap timer 166 starts a count operation. PCLCLK is continuously supplied until the count operation is completed. When the count operation is completed, a clock control signal indicating the stop of PCICLK is generated from the clock control signal output circuit 165, thereby stopping the supply of PCICLK to each of the PCI devices.

[0039]

When PCKCLK is stopped promptly when the bus becomes idle, it takes time to restart the supply of PCICLK even if an interrupt signal or the like is generated after the stop of PCKCLK, thus causing the deterioration of system performance. In this embodiment, therefore, by delaying PCICLK stop timing by a predetermined period of time by the count operation of the snap timer 166, it is possible to prevent the deterioration of the system performance and to execute as many processings as

possible collectively within a clock supply period.

[0040]

Furthermore, some PCI's require several clocks before the next event occurs after a bus transaction is completed. Due to this, by employing the snap timer 166, it is possible to promptly respond to such a bus request signal or interrupt signal from the device.

[0041]

The above description has been given only to a case of stopping PCICLK while the bus is not operating. However, if PCICLK is continuously supplied while the frequency of PCICLK is decreased instead of stopping PCICLK, it is also possible to reduce power consumption. Operation timing in this case is shown in FIG. 8.

[0042]

That is, if it is detected that both FRAME# and IRDY# are de-asserted and that a bus request signal and a system event signal are not generated, then the output of the 3-input OR gate 164 turns into "L" level indicating that the bus is not operating. As a result, the clock control signal output circuit 165 intermittently generates clock control signals so as to decrease the frequency of PCICLK, whereby the frequency of PCICLK supplied to the respective PCI devices is decreased to some fraction thereof. If a clock request signal is generated while such a slow clock is operating (i.e., if the bus monitoring circuit 161 detects the assertion of FRAME# or IRDY#, the bus request detection circuit 162 detects the occurrence of a bus

request signal or the system event detection circuit 163 detects the occurrence of an interrupt signal), the output of the 3-input OR gate 164 turns into "H" level indicating that the bus is operating. As a result, the clock control signal output circuit 165 generates a clock control signal indicating the restart of PCICLK, whereby the frequency of PCICLK supplied to the respective PCI devices is returned to an original frequency.

[0043]

Alternatively, the above-stated slow clock control and the snap timer 166 shown in FIG. 5 may be combined and the frequency of PCKCLK may be decreased a certain period of time after it is detected that the bus is not operating.

[0044]

The normal PCI devices are constituted so that part of logics in the devices can operate even while the bus clock is being stopped and an interrupt signal and a bus request signal can be normally generated, so that no problem arises. However, some PCI devices cannot operate at all if the bus clock is stopped. If such devices are connected, it is necessary to supply a minimum clock and to ensure the operation of the devices without completely stopping the bus clock. By doing so, it is possible to reduce power consumption while ensuring device operation.

[0045]

#### [Advantage of the Invention]

As stated so far, according to this invention, by detecting whether the bus is idle and detecting whether or not a bus request signal and an interrupt signal are generated, the

state of the system is examined. By controlling the stop/restart of the bus clock according to the state of the system, it is possible to stop an unnecessary bus clock without mounting special functions such as a "CLKRUN" protocol onto the PCI devices and to thereby reduce power consumption. By using the snap timer and controlling the frequency of the bus clock, in particular, it is possible to improve system performance and to realize a power saving which does not rely on the types of the PCI devices.

[Brief Description of the Drawings]

[FIG. 1] FIG. 1 is a block diagram showing the constitution of a computer system according to one embodiment of this invention.

[FIG. 2] FIG. 2 is a view showing an example of the constitution of a bus clock control circuit provided in the system in this embodiment.

[FIG. 3] FIG. 3 is a timing chart showing a clock stop operation using the bus clock control circuit shown in FIG. 2.

[FIG. 4] FIG. 4 is a timing chart showing a clock restart operation using the bus clock control circuit shown in FIG. 2.

[FIG. 5] FIG. 5 is a view showing the second example of the constitution of the bus clock control circuit provided in the system in this embodiment.

[FIG. 6] FIG. 6 is a view showing the transition of the state of the bus clock control circuit shown in FIG. 5.

[FIG. 7] FIG. 7 is a timing chart showing a clock stop operation using the bus clock control circuit shown in FIG. 5.

[FIG. 8] FIG. 8 is a timing chart showing the operation of a bus clock frequency decreasing processing realized by using the bus clock control circuit shown in FIG. 2 or 5.

[Description of Reference Symbols]

2 ... PCI bus  
3 ... ISA bus  
11 ... CPU  
12 ... host-PCI bridge  
13 ... memory  
14, 15 ... PCI peripheral device  
16 ... bus clock control circuit  
17 ... bus clock generation circuit  
18 ... PCI-DS bridge  
161 ... bus monitoring circuit  
162 ... bus request detection circuit  
163 ... system event detection circuit  
166 ... snap timer

図訳

FIG. 1

12 HOST-PCI BRIDGE;

13 MAIN MEMORY;

バスアイドル信号 BUS IDLE SIGNAL;

バス要求信号 BUS REQUEST SIGNAL;

システムイベント信号 SYSTEM EVENT SIGNAL;

16 BUS CLOCK CONTROL CIRCUIT;

クロック制御信号 CLOCK CONTROL SIGNAL;

17 BUS CLOCK GENERATION CIRCUIT;

18 PCI-ISA BRIDGE;

14 PCI DEVICE;

15 PCI DEVICE;

3 PCI BUS;

19 ISA DEVICE;

20 ISA DEVICE;

3 ISA BUS

FIG. 2

アービタ ARBITER;

割り込みコントローラ INTERRUPT CONTROLLER;

161 BUS MONITORING CIRCUIT;

162 BUS REQUEST DETECTION SIGNAL;

163 SYSTEM EVENT DETECTION CIRCUIT;

16 BUS CLOCK CONTROL CIRCUIT;

バス動作中 BUS IS BEING OPERATING;

165 OUTPUT CIRCUIT;

クロック制御信号 CLOCK CONTROL SIGNAL

FIG. 8

バスクロック BUS CLOCK ;

クロック制御信号 CLOCK CONTROL SIGNAL ;

クロック動作中 CLOCK IS BEING OPERATING ;

スロークロック動作中 SLOW CLOCK IS BEING OPERATING

FIG. 3

バスクロック BUS CLOCK ;

バス動作中 BUS IS OPERATING ;

クロック制御信号 CLOCK CONTROL SIGNAL ;

ビジー BUSY ;

アイドル IDLE ;

クロック動作中 CLOCK IS OPERATING ;

クロック停止中 CLOCK IS BEING STOPPED

FIG. 4

クロック要求信号 CLOCK REQUEST SIGNAL ;

クロック制御信号 CLOCK CONTROL SIGNAL ;

バスクロック BUS CLOCK ;

クロック停止中 CLOCK IS BEING STOPPED ;

クロック動作中 CLOCK IS BEING OPERATING

FIG. 5

アービタ ARBITER ;

割り込みコントローラ INTERRUPT CONTROLLER ;

161 BUS/MONITORING CIRCUIT;  
162 BUS REQUEST DETECTION CIRCUIT;  
163 SYSTEM EVENT DETECTION CIRCUIT;  
バス動作中 BUS IS BEING OPERATING;  
165 OUTPUT CIRCUIT;  
166 SNAP TIMER;  
クロック制御信号 CLOCK CONTROL SIGNAL;  
16 BUS CLOCK CONTROL CIRCUIT;

FIG. 6  
TIMER

FIG. 7  
バスクロック BUS CLOCK;  
バス動作中 BUS IS BEING OPERATING;  
クロック制御信号 CLOCK CONTROL SIGNAL;  
バス動作中 BUS IS BEING OPERATING;  
タイマカウント中 TIMER COUNTING;  
クロック停止中 CLOCK IS BEING STOPPED



(19)



JAPANESE PATENT OFFICE

## PATENT ABSTRACTS OF JAPAN

(11) Publication number: 11053049 A

(43) Date of publication of application: 26.02.99

(51) Int. Cl. G06F 1/04  
G06F 1/32  
G06F 1/26

(21) Application number: 09210627

(71) Applicant: TOSHIBA CORP

(22) Date of filing: 05.08.97

(72) Inventor: MAKI YASUNORI

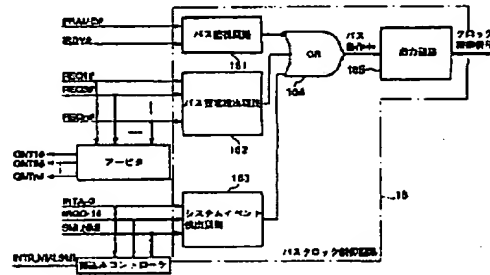
## (54) COMPUTER SYSTEM

## (57) Abstract:

**PROBLEM TO BE SOLVED:** To reduce the power consumption of a computer by enabling control over the stop and restart of a bus clock without mounting any special function on a peripheral device on a bus.

**SOLUTION:** When a bus monitor circuit 161 detects FRAME# and IRDY# being both supported and a bus request detecting circuit 162 and a system event detecting circuit 163 detect a bus request signal and a system event signal not being generated, the output of a 3-input OR gate 164 goes down to 'L' indicating that the bus is not in operation. Consequently, a clock control signal output circuit 165 generates a clock control signal indicating the stop of PCICLK, and consequently the supply to PCICLK to respective PCI devices is stopped.

COPYRIGHT: (C)1999,JPO



(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平 1 1 - 5 3 0 4 9

(43) 公開日 平成 11 年 (1999) 2 月 26 日

(51) Int. Cl. <sup>6</sup>	識別記号	F I
G 0 6 F	1/04	G 0 6 F
	1/32	1/00
	1/26	3 0 1 B
		3 3 2 Z
		3 3 4 G

審査請求 未請求 請求項の数 4

O L

(全 8 頁)

(21) 出願番号 特願平 9 - 210627

(22) 出願日 平成 9 年 (1997) 8 月 5 日

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町 72 番地

(72) 発明者 牧 康典

東京都青梅市末広町 2 丁目 9 番地 株式会社  
東芝青梅工場内

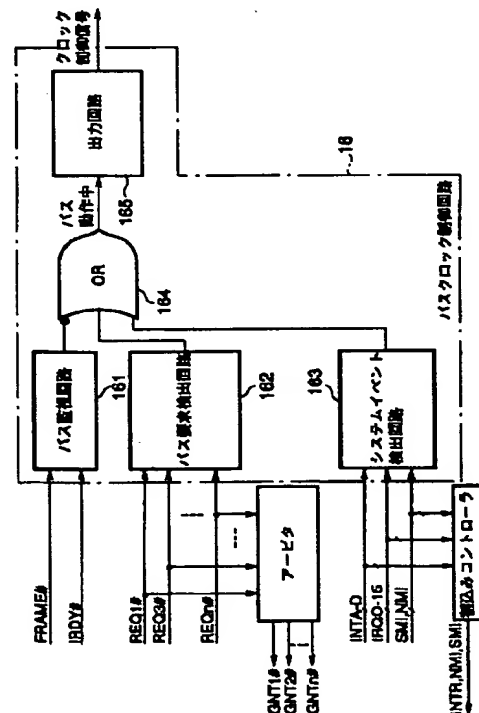
(74) 代理人 弁理士 鈴江 武彦 (外 6 名)

(54) 【発明の名称】 コンピュータシステム

(57) 【要約】

【課題】 バス上の周辺デバイスに特別な機能を実装することなくバスクロックの停止／再開を制御できるようにし、コンピュータの消費電力の低減を図る。

【解決手段】 FRAME # および IRDY # が共にデアサートされたことがバス監視回路 161 によって検出され、且つバス要求信号およびシステムイベント信号が発生されてないことがバス要求検出回路 162 およびシステムイベント検出回路 163 によって検出されると、3 入力 OR ゲート 164 の出力はバス非動作中を示す “L” レベルとなる。これにより、クロック制御信号出力回路 165 からは P C I C L K の停止を指示するクロック制御信号が発生され、これによって各 P C I デバイスに対する P C I C L K の供給が停止される。



## 【特許請求の範囲】

【請求項1】 コンピュータシステムのバスに結合される複数の周辺デバイスと、  
これら複数の周辺デバイスに前記バス上のトランザクションのタイミングを与えるためのバスクロックを生成するバスクロック生成手段と、

前記バス上のトランザクションを監視し、バスアイドル状態か否かを検出するバスアイドル検出手段と、  
前記各周辺デバイスからのバス要求信号および割り込み信号の有無を検出するイベント検出手段と、  
前記バスアイドル検出手段および前記イベント検出手段の検出結果に基づいて前記バスクロック生成手段のクロック生成動作を制御するクロック制御手段であって、バスアイドル状態であることが検出され、且つ前記バス要求信号および割り込み信号が発生されてないことが検出されたとき前記バスクロックを停止させるクロック制御手段とを具備することを特徴とするコンピュータシステム。

【請求項2】 前記クロック制御手段は、  
前記バスアイドル状態であることが検出され、且つ前記バス要求信号および割り込み信号が発生されてないことが検出されてから、一定時間経過後に前記バスクロックが停止されるように、前記バスクロックを停止させるタイミングを遅延させる手段と、  
前記バスアイドル状態の解除、前記バス要求信号、もしくは前記割り込み信号の発生が検出されたとき、前記バスクロックの供給を再開させる手段とを具備することを特徴とする請求項1記載のコンピュータシステム。

【請求項3】 コンピュータシステムのバスに結合される複数のデバイスと、  
これら複数のデバイスに前記バス上のトランザクションのタイミングを与えるためのバスクロックを生成するバスクロック生成手段と、  
前記バス上のトランザクションを監視し、バスアイドル状態か否かを検出するバスアイドル検出手段と、  
前記各デバイスからのバス要求信号および割り込み信号の有無を検出するイベント検出手段と、  
前記バスアイドル検出手段および前記イベント検出手段の検出結果に基づいて前記バスクロック生成手段のクロック生成動作を制御するクロック制御手段であって、バスアイドル状態であることが検出され、且つ前記バス要求信号および割り込み信号が発生されてないことが検出されたとき前記バスクロックの周波数を低下させるクロック制御手段とを具備することを特徴とするコンピュータシステム。

【請求項4】 前記クロック制御手段は、  
前記バスアイドル状態であることが検出され、且つ前記バス要求信号および割り込み信号が発生されてないことが検出されてから、一定時間経過後に前記バスクロックの周波数が低下されるように、前記バスクロックの周波

数を低下させるタイミングを遅延させる手段と、  
前記バスアイドル状態の解除、前記バス要求信号もしくは前記割り込み信号の発生が検出されたとき、前記バスクロックの周波数を元の状態に復帰させる手段とを具備することを特徴とする請求項3記載のコンピュータシステム。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】この発明はコンピュータシステムに関し、特にバスクロック制御機能を有するコンピュータシステムに関する。

## 【0002】

【従来の技術】従来、パーソナルコンピュータに使用されるシステムバスとしては、ISA (Industry Standard Architecture) バスやEISA (Extended ISA) バスが主流であったが、最近では、データ転送速度の高速化や、プロセッサに依存しないシステムアーキテクチャの構築のために、PCI (Peripheral Component Interconnect) バスが採用され始めている。

【0003】PCIバスにおいては、ほとんど全てのデータ転送はブロック転送を基本としており、これら各ブロック転送はバースト転送を用いて実現されている。これにより、PCIバスでは最大133Mバイト/秒（データバスが32ビット幅の時）のデータ転送速度を実現できる。

【0004】したがって、PCIバスを採用すると、I/Oデバイス間、およびシステムメモリとI/Oデバイスとの間のデータ転送などを高速に行うことが可能となり、システム性能を高めることができる。

【0005】このようなPCIバスアーキテクチャでは、高速動作の実現のために33MHzという比較的高速なバスクロックが用いられている。このバスクロックはPCIバス上のトランザクションのタイミングをバス上の各周辺デバイスに提供するためのものであり、システム動作中は常にバス上の周辺デバイスに供給され続けている。

## 【0006】

【発明が解決しようとする課題】しかし、このような高速バスクロックは、コンピュータの電力消費量を高める大きな要因の一つとなる。このため、最近では、PCIバスの仕様の中にもバスクロックを停止させるための仕様（“CLKRUN”プロトコル）が盛り込まれるようになってきている。

【0007】ところが、この“CLKRUN”プロトコルをシステムに実装するためには、PCIバス上の全てのデバイスが“CLKRUN”プロトコルに対応するための機能を持つことが必要とされる。このため、“CLKRUN”プロトコルに対応してない既存の周辺デバイ

スが存在する場合には“CLKRUN”プロトコルを用いることができず、バスクロックを停止させることはできなかった。

【0008】この発明はこのような点に鑑みてなされたものであり、周辺デバイスに“CLKRUN”プロトコルなどの特別な機能を実装することなくバスクロックを制御できるようにし、容易に消費電力の低減を図ることが可能なコンピュータシステムを提供することを目的とする。

【0009】

【課題を解決するための手段】この発明によるコンピュータシステムは、コンピュータシステムのバスに結合される複数の周辺デバイスと、これら複数の周辺デバイスに前記バス上のトランザクションのタイミングを与えるためのバスクロックを生成するバスクロック生成手段と、前記バス上のトランザクションを監視し、バスアイドル状態か否かを検出するバスアイドル検出手段と、前記各周辺デバイスからのバス要求信号および割り込み信号の有無を検出するイベント検出手段と、前記バスアイドル検出手段および前記イベント検出手段の検出結果に基づいて前記バスクロック生成手段のクロック生成動作を制御するクロック制御手段であって、バスアイドル状態であることが検出され、且つ前記バス要求信号および割り込み信号が発生されてないことが検出されたとき前記バスクロックを停止させるクロック制御手段とを具備することを特徴とする。

【0010】このコンピュータシステムにおいては、バスアイドル状態、バス要求信号および割り込み信号の有無を検出することによりシステム状態が調べられ、バスアイドル状態で、且つバス要求信号および割り込み信号が発生されてないことが検出されたときには、システムアイドルであると判断されてバスクロックが停止される。この場合、たとえバスアイドル状態であってもバス要求信号または割り込み信号が発生されている限りにおいてはバスクロックは停止されないで、周辺デバイスの動作に影響を与えることはない。よって、周辺デバイスに“CLKRUN”プロトコルなどの特別な機能を実装することなく不要なバスクロックを停止できるようになり、電力消費の低減を図ることができる。

【0011】また、前記クロック制御手段としては、前記バスアイドル状態であることが検出され、且つ前記バス要求信号および割り込み信号が発生されてないことが検出されてから一定時間経過後に前記バスクロックが停止されるように、前記バスクロックを停止させるタイミングを遅延させる手段を含むことが好ましい。

【0012】システムアイドル時にすぐにバスクロックを停止させると、その後に割り込み信号などが発生されてもバスクロックの供給再開までには時間を要するので、システムパフォーマンスの低下を招くことになる。従って、システムパフォーマンスの低下を防止するため

には、クロック供給期間になるべく多くの処理をまとめて実行させるようにすることが必要であり、これはバスクロックを停止させるタイミングを遅延させることによって実現できる。

【0013】また、この発明のコンピュータシステムは、コンピュータシステムのバスに結合される複数のデバイスと、これら複数のデバイスに前記バス上のトランザクションのタイミングを与えるためのバスクロックを生成するバスクロック生成手段と、前記バス上のトランザクションを監視し、バスアイドル状態か否かを検出するバスアイドル検出手段と、前記各デバイスからのバス要求信号および割り込み信号の有無を検出するイベント検出手段と、前記バスアイドル検出手段および前記イベント検出手段の検出結果に基づいて前記バスクロック生成手段のクロック生成動作を制御するクロック制御手段であって、バスアイドル状態であることが検出され、且つ前記バス要求信号および割り込み信号が発生されてないことが検出されたとき前記バスクロックの周波数を低下させるクロック制御手段とを具備することを特徴とする。

【0014】この構成によれば、バスクロックを完全に停止する代わりに、そのバスクロックの周波数が低下される。通常、周辺デバイスは、バスクロックが停止されている状態においてもその内部の一部のロジックについては動作できるように構成されており割り込み信号やバス要求信号を正常に発生できるので問題はないが、周辺デバイスによってはバスクロックが停止してしまうと全く動作できないものもある。したがって、このようなデバイスが接続されている場合には、バスクロックを完全に停止するのではなく、必要最低限のクロックを供給してそれらデバイスの動作を保証することが必要である。これにより、デバイスの動作を保証しつつ、電力消費の低減を図ることが可能となる。

【0015】

【発明の実施の形態】以下、図面を参照してこの発明の実施形態を説明する。図1には、この発明の一実施形態に係わるコンピュータシステムの構成が示されている。このコンピュータシステムは、バッテリー駆動可能なノートブックタイプまたはラップトップタイプのポータブルコンピュータであり、そのシステムボード上には、プロセッサバス1、PCIバス2、ISAバス3、CPU11、ホストPCIブリッジ装置12、主メモリ13、各種PCI周辺デバイス14、15、バスクロック制御回路16、バスクロック生成回路17、PCI-ISAブリッジ装置18、および各種ISA周辺デバイス19、20などが設けられている。

【0016】これらコンポーネントの内、PCIバス2に接続されたPCIデバイスとして機能するもの、すなわちホストPCIブリッジ装置12、各種PCI周辺デバイス14、15、PCI-ISAブリッジ装置18

には、バスクロック生成回路 17 によって生成された P C I バスクロック (P C I C L K) が供給される。この P C I バスクロック (P C I C L K) は、各 P C I デバイスに対してバストランザクションのタイミングを与えるものであり、P C I バス 2 上のサイクルは全て P C I バスクロック (P C I C L K) に同期して実行される。

【0017】次に、図 1 の各コンポーネントの機能および構成について説明する。C P U 11 は、例えば、米インテル社によって製造販売されているマイクロプロセッサ “P e n t i u m” などによって実現されている。この C P U 11 の入出力ピンに直結されているプロセッサバス 1 は、64 ビット幅のデータバスを有している。主メモリ 13 は、オペレーティングシステム、デバイスドライバ、実行対象のアプリケーションプログラム、および処理データなどを格納するメモリデバイスであり、複数の D R A M モジュールによって構成されている。

【0018】ホスト P C I ブリッジ装置 12 は、プロセッサバス 1 と P C I バス 2 との間を繋ぐブリッジ L S I であり、P C I バス 2 のバスマスタの 1 つとして機能する。このホスト P C I ブリッジ装置 12 は、プロセッサバス 1 と P C I バス 2 との間で、データおよびアドレスを含むバスサイクルを双方向で変換する機能、およびメモリバスを介して主メモリ 13 をアクセス制御する機能などを有している。

【0019】P C I バス 2 はクロック同期型の入出力バスであり、前述したように P C I バス 2 上の全てのサイクルは P C I バスクロックに同期して行なわれる。P C I バスクロックの周波数は最大 33 M H z である。P C I バス 2 は、時分割的に使用されるアドレス/データバスを有している。このアドレス/データバスは、32 ビット幅である。

【0020】P C I バス 2 上のデータ転送サイクルは、アドレスフェーズとそれに後続する 1 以上のデータフェーズとから構成される。アドレスフェーズにおいてはアドレスおよび転送タイプが出力され、データフェーズでは 8 ビット、16 ビット、24 ビットまたは 32 ビットのデータが出力される。

【0021】P C I デバイス 14、15 は例えばグラフィックスコントローラ、P C カード (カードバス) コントローラ、I r D A コントローラ、S C S I コントローラなどであり、ホスト P C I ブリッジ装置 12 と同様に P C I バス 2 のバスマスタとして機能する。

【0022】P C I - I S A ブリッジ装置 16 は、P C I バス 2 と I S A バス 3 との間を繋ぐブリッジ L S I であり、P C I デバイスの 1 つとして機能する。I S A バス 3 上の I S A デバイス 19、20 は、例えば、H D D、システムタイマ、キーボードコントローラなどである。

【0023】バスクロック制御回路 16 は各 P C I デバイスに対する P C I バスクロック (P C I C L K) の供

給の停止/再開を制御するためのものであり、その P C I バスクロック (P C I C L K) の停止/再開の制御は、バスアイドル信号、バス要求信号、およびシステムイベント信号に基づいて行われる。バスアイドル信号は P C I バス 2 の動作状態、つまりバス動作中 (バスサイクル実行中) であるか否かを示す。バス要求信号は、バスマスタが P C I バス 2 の使用权を P C I バスアービタに要求するための信号である。システムイベント信号は、割り込み信号などシステム内で何らかのイベントが発生したことを示すものである。

【0024】バスクロック制御回路 16 は、バスアイドル信号による P C I バスのアイドルを検出し、バス要求信号、およびシステムイベント信号がなければ、クロック制御信号によりバスクロック生成回路 17 を制御してバスクロックを停止させる。また、バスクロック停止中に、バス要求信号またはシステムイベント信号が発生した場合には、バスクロック制御回路 16 は、クロック制御信号によりバスクロック生成回路 17 を制御してバスクロックの供給を再開させる。

【0025】図 2 には、バスクロック制御回路 16 の具体的な構成例が示されている。図示のように、バスクロック制御回路 16 は、バス監視回路 161、バス要求検出回路 162、システムイベント検出回路 163、O R ゲート 164、およびクロック制御信号出力回路 165 から構成されている。

【0026】バス監視回路 161 は前述のバスアイドル信号を生成するためのものであり、P C I バス 2 上に定義されたフレーム信号 F R A M E # およびイニシエータレディー信号 I R D Y # を用いて P C I バス 2 上のトラランザクションを監視し、バス非動作であることを検出したときにバスアイドル信号を “H” レベルにアサートし、バス動作中であることを検出したときにはバスアイドル信号を “L” レベルにデアサートする。

【0027】ここで、F R A M E # は、トラランザクションの開始とその期間を示すために現在のマスタによってドライブされる信号である。F R A M E # がデアサートされた時、トラランザクションが最後のデータフェーズであることを示す。I R D Y # は現在のマスタによってドライブされる信号であり、ライトサイクルにおいては、マスタがバス上に確定データを出力したことを示すためにアサートされ、リードサイクルにおいては、マスタがデータを受信する準備ができたことを示すためにアサートされる。

【0028】これら F R A M E # および I R D Y # が共にデアサートされているとき、バス監視回路 161 はバスアイドル状態であることを検出し、バスアイドル信号を “H” レベルにアサートする。

【0029】バス要求検出回路 162 は、P C I デバイスそれぞれから P C I バスアービタに入力される全てのバスリクエスト信号 R E Q 1 # ~ R E Q n # を監視し、

REQ1#~REQn#のいずれかがアサートされているときバス要求信号を“H”レベルにアサートする。

【0030】システムイベント検出回路は、各PCIデバイスから割り込みコントローラに入力される割り込み信号INTA-D、各ISAデバイスから割り込みコントローラに入力される割り込み信号IRQ0-15、さらにマスク不能割り込み信号NMI、システム管理割り込み信号SMIについての監視を行い、いずれかの割り込み信号が発生されているときシステムイベント信号を“H”レベルにアサートする。

【0031】バス監視回路161からのバスアイドル信号は、3入力ORゲート164の第1入力に反転入力される。また、バス要求検出回路162およびシステムイベント検出回路163からのバス要求信号およびシステムイベント信号は、そのまま3入力ORゲート164の第2入力および第3入力にそれぞれ入力される。3入力ORゲート164の出力はバス動作中(“H”レベル)／バス非動作中(“L”レベル)を示すものであり、クロック制御信号出力回路165に入力される。

【0032】クロック制御信号出力回路165は、3入力ORゲート164の出力に基づいてPCICLKの停止／再開を指示するクロック制御信号を発生する。次に、図3および図4を参照して、本実施形態の動作を説明する。

【0033】図3のタイミングチャートはバスクロック(PCICLK)を停止させる場合のタイミングである。すなわち、FRAME#およびIRDY#が共にデアサートされたことが検出され、且つバス要求信号およびシステムイベント信号が発生されてないことが検出されると、3入力ORゲート164の出力はバス非動作中を示す“L”レベルとなる。これにより、クロック制御信号出力回路165からはPCICLKの停止を指示するクロック制御信号が発生され、これによって各PCIデバイスに対するPCICLKの供給が停止される。

【0034】このPCICLKの供給停止期間中に、クロック要求信号が発生されると(バス監視回路161によるFRAME#またはIRDY#のアサートの検出、バス要求検出回路162によるバス要求信号の発生の検出、またはシステムイベント検出回路163による割り込み信号の発生の検出)、3入力ORゲート164の出力はバス動作中を示す“H”レベルとなる。これにより、クロック制御信号出力回路165からはPCICLKの再開を指示するクロック制御信号が発生され、これによって各PCIデバイスに対するPCICLKの供給が再開される。

【0035】このように、図2の構成によれば、バスアイドル状態、バス要求信号および割り込み信号の有無を検出することによりシステム状態が調べられ、バスアイドル状態で、且つバス要求信号および割り込み信号が発生されてないことが検出されたときには、システムアイ

ドルであると判断されてPCICLKが停止される。この場合、たとえバスアイドル状態であってもバス要求信号または割り込み信号が発生されている限りにおいてはPCICLKは停止されないで、各PCIデバイスの動作に影響を与えることはない。また、PCICLKの停止中にPCIデバイスからバス要求信号や割り込み信号が発生されると、PCICLKの供給が再開され、PCICLKに同期した正常なバストランザクションを行うことが可能となる。よって、PCIデバイスに“CLKRUN”プロトコルなどの特別な機能を実装することなく不要なバスクロックを停止できるようになり、電力消費の低減を図ることができる。

【0036】図5には、バスクロック制御回路16の第2の構成例が示されている。ここでは、図2の構成に加え、スナップタイマ166が設けられている。このスナップタイマ166は、PCICLKの停止タイミングを一定期間遅らせるためのものであり、3入力ORゲート164の出力がバス非動作中を示す“L”レベルとなつてからカウント動作を開始し、所定のカウンタ値に達したときにそれをクロック制御信号出力回路165に通知する。これにより、PCICLKの停止を指示するクロック制御信号は、スナップタイマ166のカウント時間だけ遅れて発生されることになる。スナップタイマ166のカウント値は、たとえばバスクロック制御回路16内にカウンタ値を設定するためのコンフィグレーションレジスタを設け、そこに希望する遅れ時間に相当するカウンタ値をソフトウェア的に設定することによって、プログラマブルにすることができる。

【0037】図6には、図5のバスクロック制御回路16の状態遷移の様子が示されている。図6において、状態S1(RUN)はPCICLKが供給されている状態を示し、この状態でバス非動作が検知されると、状態S2(SNAP)に移行される。状態S2(SNAP)では、スナップタイマ166のカウント動作が行われる。スナップタイマ166のカウント動作中に割り込み信号の発生やバス要求信号の発生が検出されると、状態S1(RUN)に復帰されると共に、スナップタイマ166のカウント値は初期値に戻される。一方、状態S2(SNAP)においてスナップタイマ166のカウント動作が完了すると、状態S3(STOP)に移行される。状態S3(STOP)では、PCICLKの供給が停止される。状態S3(STOP)において割り込み信号の発生やバス要求信号の発生が検出されると、状態S1(RUN)に復帰される。

【0038】図7には、状態S2(SNAP)経由で状態S1(RUN)から状態S3(STOP)に遷移するときの動作タイミングが示されている。すなわち、FRAME#およびIRDY#が共にデアサートされたことが検出され、且つバス要求信号およびシステムイベント信号が発生されてないことが検出されると、3入力OR

10

20

30

40

50

ゲート164の出力はバス非動作中を示す“L”レベルとなる。これにより、スナップタイマ166のカウント動作が開始される。そのカウント動作が完了するまではPCICLKは供給され続ける。そして、カウント動作が完了すると、クロック制御信号出力回路165からPCICLKの停止を指示するクロック制御信号が発生され、これによって各PCIデバイスに対するPCICLKの供給が停止される。

【0039】バスアイドル時にすぐにPCICLKを停止させると、その後に割り込み信号などが発生されてもPCICLKの供給再開までには時間を要するので、システムパフォーマンスの低下を招くことになる。従って、本例のように、PCICLKの停止タイミングをスナップタイマ166のカウント動作によって一定期間遅延させることにより、システムパフォーマンスの低下を防止できるようになり、クロック供給期間内になるべく多くの処理をまとめて実行させることが可能となる。

【0040】また、PCIによってはバストランザクションが完了してから、つぎのイベントを起こすまでに何クロックか必要なものがあるので、スナップタイマ166を使用することによって、そのようなデバイスからのバス要求信号や割り込み信号に即座に 응답できるようにする。

【0041】なお、以上の説明では、バス非動作中にPCICLKを停止する場合の例についてのみ説明したが、PCICLKを停止する代わりに、その周波数を低下させた状態でPCICLKを供給し続けるようにしても電力消費を低減することができる。この場合における動作タイミングを図8に示す。

【0042】すなわち、FRAME#およびIRDY#が共にデアサートされたことが検出され、且つバス要求信号およびシステムイベント信号が発生されてないことが検出されると、3入力ORゲート164の出力はバス非動作中を示す“L”レベルとなる。これにより、クロック制御信号出力回路165からはPCICLKの周波数を低下させるためにクロック制御信号が断続的に発生され、これによって各PCIデバイスに供給されるPCICLKの周波数は何分の1かに低減される。このようなスロークロック動作中に、クロック要求信号が発生されると（バス監視回路161によるFRAME#またはIRDY#のアサートの検出、バス要求検出回路162によるバス要求信号の発生の検出、またはシステムイベント検出回路163による割り込み信号の発生の検出）、3入力ORゲート164の出力はバス動作中を示す“H”レベルとなる。これにより、クロック制御信号出力回路165からはPCICLKの再開を指示するクロック制御信号が発生され、これによって各PCIデバイスに供給されるPCICLKの周波数は元の周波数に戻される。

【0043】また、このようなスロークロック制御と図

5のスナップタイマ166とを組み合わせ使用し、バス非動作中が検出されてから一定時間経過後にPCICLKの周波数を落とすようにしてもよい。

【0044】通常のPCIデバイスは、バスクロックが停止されている状態においてもその内部の一部のロジックについては動作できるように構成されており割り込み信号やバス要求信号を正常に発生できるので問題はないが、PCIデバイスによってはバスクロックが停止されてしまうと全く動作できないものもある。したがって、このようなデバイスが接続されている場合には、前述したように、バスクロックを完全に停止するのではなく、必要最低限のクロックを供給してそれらデバイスの動作を保証することが好ましい。これにより、デバイスの動作を保証しつつ、電力消費の低減を図ることが可能となる。

#### 【0045】

【発明の効果】以上説明したように、この発明によれば、バスアイドル状態、バス要求信号および割り込み信号の有無を検出することによりシステム状態を調べ、それに応じてバスクロックの停止／再開を制御することにより、PCIデバイスに“CLKRUN”プロトコルなどの特別な機能を実装することなく不要なバスクロックを停止できるようになり、電力消費の低減を図ることができる。特に、スナップタイマの使用およびバスクロック周波数の制御を行うことにより、システムパフォーマンスの向上やPCIデバイスの種類に依存しないパワーセーブを実現できるようになる。

#### 【図面の簡単な説明】

【図1】この発明の一実施形態に係るコンピュータシステムの構成を示すブロック図。

【図2】同実施形態のシステムに設けられたバスクロック制御回路の構成の一例を示す図。

【図3】図2のバスクロック制御回路を用いたクロック停止動作を示すタイミングチャート。

【図4】図2のバスクロック制御回路を用いたクロック再開動作を示すタイミングチャート。

【図5】同実施形態のシステムに設けられたバスクロック制御回路の第2の構成例を示す図。

【図6】図5のバスクロック制御回路の状態遷移を示す図。

【図7】図5のバスクロック制御回路を用いたクロック停止動作を示すタイミングチャート。

【図8】図2または図5のバスクロック制御回路を用いて実現されるバスクロック周波数低下処理の動作を示すタイミングチャート。

#### 【符号の説明】

2…PCIバス

3…ISAバス

11…CPU

12…ホスト-PCIブリッジ

11

12

13…メモリ

14, 15…PCI周辺デバイス

16…バスクロック制御回路

17…バスクロック生成回路

18…PCI-DSブリッジ

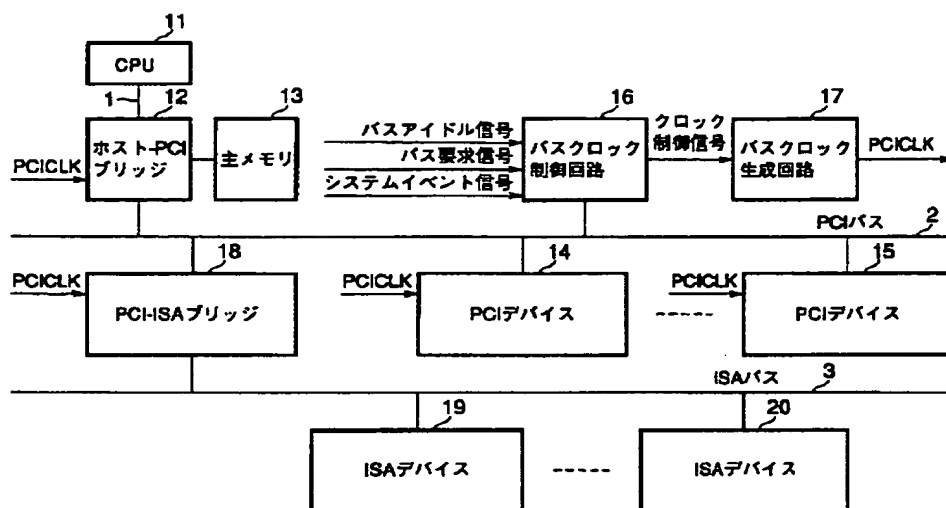
161…バス監視回路

162…バス要求検出回路

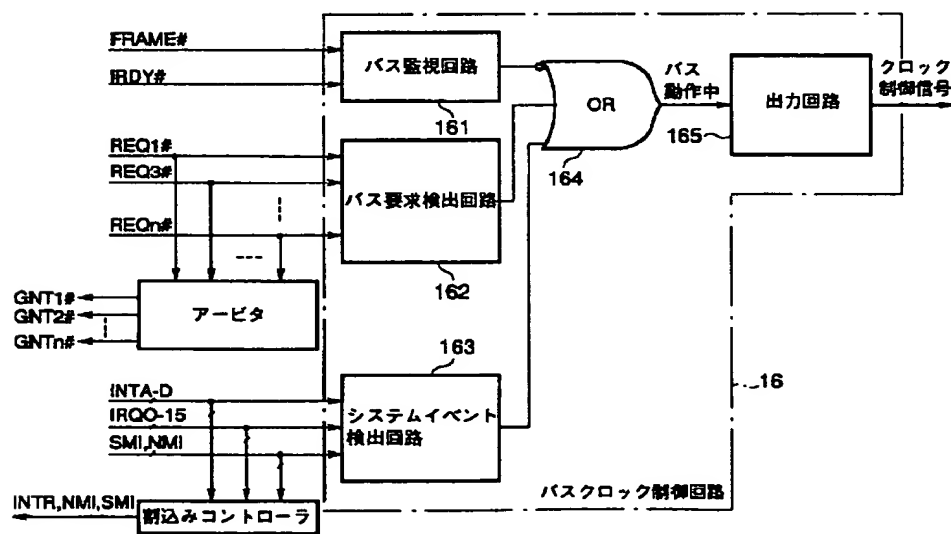
163…システムイベント検出回路

166…スナップタイマ

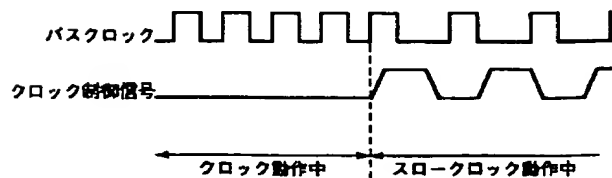
【図1】



【図2】

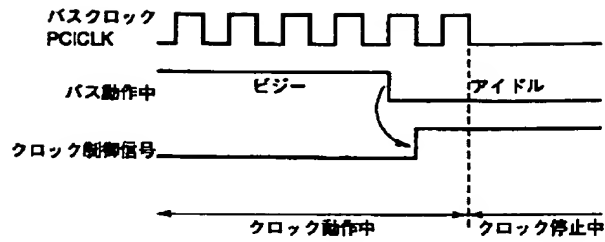


【図8】

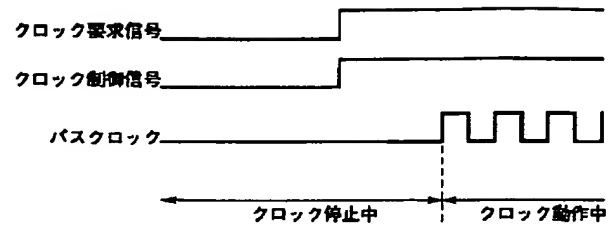




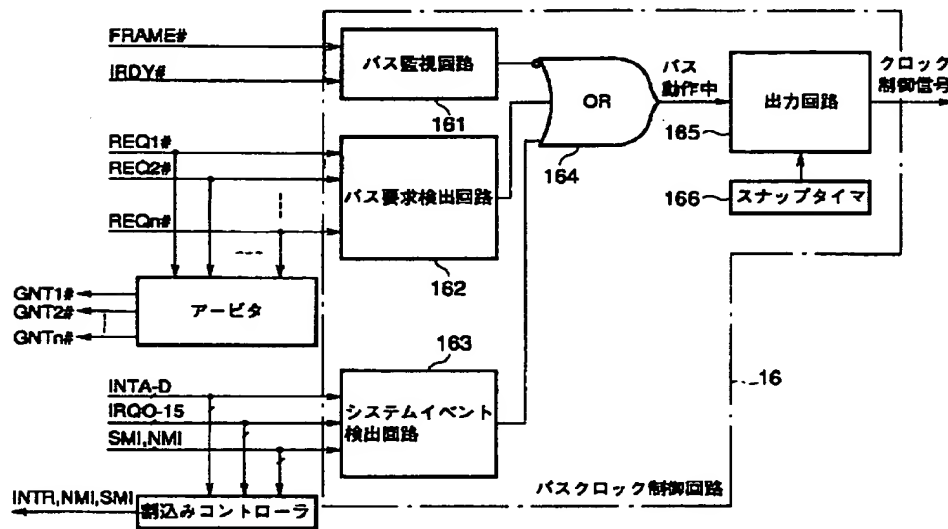
【図 3】



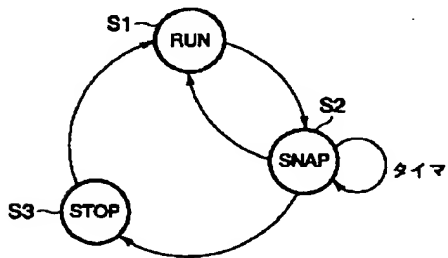
【図 4】



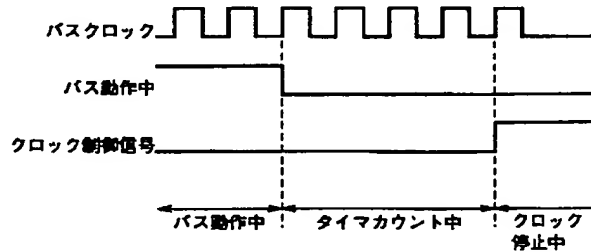
【図 5】



【図 6】



【図 7】



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**